Adrien Protzel

# Concurrency Control

1. Consider the following classes of schedules: serializable and 2PL. For each of the following schedules, state which of the preceding classes it belongs to. If you cannot decide whether a schedule belongs in a certain class based on the listed actions, explain briefly. Also, for each 2PL schedule, identify whether a cascading rollback (abort) may happen. A cascading rollback will happen in a schedule if a given transaction aborts at some point in the schedule, and at least one other transaction must be aborted by the system to keep the database consistent.

   The actions are listed in the order they are scheduled and prefixed with the transaction name. If a commit or abort is not shown, the schedule is incomplete; assume that abort or commit must follow all the listed actions.

a. T1:R(X), T2:R(Y), T3:W(X), T2:R(X), T1:R(Y)
   Serializable: Yes
   2PL: Yes
   Cascade: Possible: T2 reads X after T3, before T3 commits

b. T1:R(X), T1:R(Y), T1:W(X), T2:R(Y), T3:W(Y), T1:W(X), T2:R(Y)
   Serializable: No: Conflict of T2 and T3 on Y, read > write > read of Y
   2PL: Yes
   Cascade: Possible: T2 reads Y before and after T3 writes

c. T1:W(X), T2:R(X), T1:W(X)
   Serializable: No: Conflict of T1 and T2 on X
   2PL: No: T1 and T2 access X at the same time
   Cascade: Possible: T2 reads X before T1 commit

d. T1:R(X), T2:W(X), T1:W(X), T3:R(X)
   Serializable: No: Conflict of T1 and T2 on X
   2PL: No: T1 writes after T2 writes
   Cascade: Possible: T1 and T3 reads before T2 commit

e. T1:R(X), T2:W(X), T3:R(X), T1:R(Y)
   Serializable: Yes
   2PL: Yes
   Cascade: Possible: T3 reads X before T2 commit

2. Consider the following schedule:

| | T1 | T2 |
|---|---|---|
| 0 | start | |
| 1 | Read x | |
| 2 | Write x | |
| 3 | | start |
| 4 | | Read x |
| 5 | | Write x |
| 6 | | commit |
| 7 | Read y | |
| 8 | Write y | |
| 9 | commit | |

T1(S) > T1(RX) > T1(WX) > T2(S) > T2(RX) > T2(WX) > T2(C) > T1(RY) > T1(WY) > T1(C)

What are the maximum degrees of consistency for T1 and T2 in this schedule? You must find the maximum degrees of consistency for T1 and T2 that make this schedule possible (2 points).

> In the schedule, T2 reads and writes X after T1 modifies it but before T1 commits, resulting in a dirty read. This means T2's maximum consistency level is Read Uncommitted. T1, however, does not read any uncommitted data from other transactions, so its maximum consistency level is Read Committed.